

Inhaltsverzeichnis

1	Einleitung	1
1.1	Releasepolitik	1
1.2	Inhaltsübersicht: Was erwartet Sie im Folgenden?	4
1.3	Grundgerüst des Eclipse-Projekts	5
1.4	Anmerkung zum Programmierstil	6
1.4.1	Gedanken zur Sourcecode-Kompaktheit	6
1.4.2	Gedanken zu <code>final</code> und <code>var</code>	7
1.4.3	Blockkommentare in Listings	8
1.4.4	Gedanken zur Formatierung	8
1.5	Konfigurationen für Build-Tools und IDEs	9
1.5.1	Java 17 mit Gradle	9
1.5.2	Java 17 mit Maven	10
1.5.3	Java 17 mit Eclipse	11
1.5.4	Java 17 mit IntelliJ	12
1.6	Ausprobieren der Beispiele und Lösungen	13
1.6.1	Ausprobieren neuer Java-Features mit der JShell oder der Kommandozeile	13
1.6.2	Ausprobieren neuer Java-Features in einer Sandbox	16
I	Neuerungen in Java 11 bis 17	17
2	Neuerungen in Java 17 im Überblick	19
2.1	JEPs im Überblick	19
2.1.1	JEP 306: Restore Always-Strict Floating-Point Semantics ...	20
2.1.2	JEP 356: Enhanced Pseudo-Random Number Generators ..	20
2.1.3	JEP 382: New macOS Rendering Pipeline	21
2.1.4	JEP 391: macOS/AArch64 Port	21
2.1.5	JEP 398: Deprecate the Applet API for Removal	22
2.1.6	JEP 403: Strongly Encapsulate JDK Internals	22
2.1.7	JEP 406: Pattern Matching for <code>switch</code> (Preview)	23
2.1.8	JEP 407: Remove RMI Activation	23
2.1.9	JEP 409: Sealed Classes	24

2.1.10	JEP 410: Remove the Experimental AOT and JIT Compiler . . .	24
2.1.11	JEP 411: Deprecate the Security Manager for Removal	24
2.1.12	JEP 412: Foreign Function & Memory API (Incubator)	25
2.1.13	JEP 414: Vector API (Second Incubator)	25
2.1.14	JEP 415: Context-Specific Deserialization Filters	25
2.2	Neue Releasepolitik im Überblick	26
3	Syntaxneuerungen bis Java 17	27
3.1	Text Blocks	28
3.1.1	Grundlegende Syntax	28
3.1.2	Besonderheiten	30
3.1.3	Platzhalter	33
3.2	Switch Expressions	34
3.2.1	Einführendes Beispiel	34
3.2.2	Vollständigkeitsprüfung	36
3.2.3	Fallstricke der alten Syntax und Abhilfen	37
3.2.4	Rückgabe mit <code>yield</code>	39
3.2.5	Rückwärtskompatible Angabe bei <code>case</code> mit <code>yield</code>	39
3.3	Records	41
3.3.1	Einführendes Beispiel	41
3.3.2	Records für DTOs und Rückgabewerte	43
3.3.3	Erweiterungsmöglichkeiten	45
3.3.4	Besonderheit – Generics in Records	48
3.3.5	Besonderheit – Records und Interfaces	50
3.3.6	Zusammenfassung	50
3.4	Pattern Matching bei <code>instanceof</code>	52
3.4.1	Neue Syntax und einführendes Beispiel	52
3.4.2	Im Praxiseinsatz	53
3.5	Sealed Types	54
3.5.1	Neue Schlüsselwörter	54
3.5.2	Einführendes Beispiel	55
3.5.3	Wissenswertes	57
3.6	Lokale Enums und Interfaces	58
3.7	Statische Attribute und Methoden in inneren Klassen	59
3.8	Pattern Matching und Erweiterung für <code>switch</code> (Preview)	59
3.8.1	Einfaches Pattern Matching	60
3.8.2	Pattern Matching mit Bedingung	61
3.8.3	Spezialfall: Behandlung von <code>null</code>	62
4	Übungen zu den Syntaxneuerungen in JDK 11 bis 17	63
4.1	Aufgaben	63
4.2	Lösungen	69

5	Neues und Änderungen in den Java-17-APIs	81
5.1	Erweiterungen in der Klasse <code>String</code>	81
5.1.1	Die Methode <code>isBlank()</code>	82
5.1.2	Die Methode <code>lines()</code>	82
5.1.3	Die Methode <code>repeat(int)</code>	83
5.1.4	Die Methoden <code>strip()</code> , <code>stripLeading()</code> und <code>stripTrailing()</code>	84
5.1.5	Die Methode <code>indent()</code>	85
5.1.6	Die Methode <code>transform()</code>	86
5.1.7	Die Methode <code>formatted()</code>	87
5.2	Erweiterung im Interface <code>Predicate<T></code>	88
5.3	Erweiterung in der Klasse <code>Optional<T></code>	89
5.4	Erweiterungen in der Utility-Klasse <code>Files</code>	90
5.4.1	Die Methoden <code>writeString()</code> und <code>readString()</code>	90
5.4.2	Die Methode <code>mismatch()</code>	91
5.5	Erweiterungen im Stream-API	93
5.5.1	Der <code>teeing()</code> -Kollektor	93
5.5.2	Die Methode <code>toList()</code> als Shortcut zum Kollektor	96
5.5.3	Die Intermediate Operation <code>mapMulti()</code>	96
5.6	HTTP/2-API	100
5.6.1	Einführung	100
5.6.2	Real-World-Example: Wechselkurs mit REST	103
5.7	Die Utility-Klasse <code>CompactNumberFormat</code>	105
5.7.1	Einführendes Beispiel	105
5.7.2	Rundung steuern	108
5.7.3	Eigene Einheiten angeben	109
5.8	Verschiedenes und Deprecations	111
5.8.1	Unix-Domain-Socket Channels	111
5.8.2	Day Period Support	115
5.8.3	Aufruf von Defaultmethoden aus Dynamic Proxies	117
5.8.4	Deprecations der Konstruktoren der Wrapper-Klassen	120
6	Übungen zu den API-Neuerungen in JDK 11 bis 17	123
6.1	Aufgaben	123
6.2	Lösungen	129
7	Änderungen in der JVM bis Java 17	139
7.1	Verbesserung bei <code>NullPointerExceptions</code>	140
7.1.1	Einführende Beispiele	140
7.1.2	Fehlersuche bei komplexen Ausdrücken	142
7.2	Java + REPL => <code>jshell</code>	144
7.2.1	Einführendes Beispiel	144
7.2.2	Weitere Kommandos und Möglichkeiten	145

7.2.3	Neuere Java-Features nutzen	147
7.2.4	Komplexere Aktionen	148
7.2.5	JShell-API	151
7.3	Launch Single-File Source-Code Programs (JDK 11)	153
7.3.1	Einführendes Beispiel	153
7.3.2	Palindrom-Prüfung	154
7.3.3	REST-Calls mit HTTP/2	155
7.3.4	Besonderheit: Shebang-Skript	155
7.3.5	Besonderheit: Preview-Features ausprobieren	158
7.4	Microbenchmarks und JMH	159
7.4.1	Eigene Microbenchmarks und Varianten davon	159
7.4.2	Microbenchmarks mit JMH	162
7.4.3	Weitere Microbenchmarks mit JMH	169
7.4.4	Fallstricke beim Benchmarking mit JMH	172
7.4.5	Projekt zum Experimentieren	176
7.4.6	Fazit zu JMH	177
7.5	Das Packaging-Tool <code>jpckage</code>	177
7.5.1	Einführung	177
7.5.2	<code>jpckage</code> am Beispiel	178
7.5.3	Externe Bibliotheken mit <code>jpckage</code> einbinden	181
7.5.4	Hintergrundwissen: Verzeichnisstruktur und -inhalt der Distributionsen	185
7.6	Veränderungen bei den Garbage Collectors	186
7.6.1	Epsilon Garbage Collector (JDK 11)	186
7.6.2	Der Garbage Collector namens ZGC (JDK 15)	187
7.6.3	Entfernung von Concurrent Mark Sweep (CMS)	187
7.7	Ausgliederungen/Deprecations	187
7.7.1	Entfernung der JavaScript-Engine	187
7.7.2	Das Tool <code>jdeprscan</code>	189
7.7.3	Ausgliederung von JavaFX	192
7.7.4	Ausgliederung von Java EE und CORBA	192
7.7.5	Beispiel JAXB	193
8	Übungen zu den JVM-Neuerungen in JDK 11 bis 17	199
8.1	Aufgaben	199
8.2	Lösungen	201

II Ausblick	203
9 Neuerungen in Java 18	205
9.1 Java-18-JEPs im Überblick	205
9.1.1 JEP 400: UTF-8 by Default	206
9.1.2 JEP 408: Simple Web Server	207
9.1.3 JEP 413: Code Snippets in Java API Documentation	209
9.1.4 JEP 416: Reimplement Core Reflection with Method Handles	210
9.1.5 JEP 417: Vector API (Third Incubator)	212
9.1.6 JEP 418: Internet-Address Resolution SPI	213
9.1.7 JEP 419: Foreign Function & Memory API (Second Incubator)	215
9.1.8 JEP 420: Pattern Matching for <code>switch</code> (Second Preview)	216
9.1.9 JEP 421: Deprecate Finalization for Removal	220
9.2 API-Neuerungen	220
9.2.1 Neuerungen in der Klasse <code>FileInputStream</code>	220
9.2.2 Neuerungen in der Klasse <code>Math</code>	220
9.2.3 Neuerungen in der Klasse <code>Duration</code>	221
9.2.4 Neuerungen in der Aufzählung <code>SourceVersion</code>	222
9.2.5 Deprecations	222
9.3 Notwendige Anpassungen für Build-Tools und IDEs	223
9.3.1 Java 18 mit Gradle	224
9.3.2 Java 18 mit Maven	225
9.3.3 Java 18 mit Eclipse	226
9.3.4 Java 18 mit IntelliJ	227
9.4 Fazit	228
10 Ausblick auf Java 19	229
10.1 Java-19-JEPs im Überblick	230
10.1.1 JEP 405: Record Patterns (Preview)	231
10.1.2 JEP 422: Linux/RISC-V Port	233
10.1.3 JEP 424: Foreign Function & Memory API (Preview)	233
10.1.4 JEP 425: Virtual Threads (Preview)	234
10.1.5 JEP 426: Vector API (Fourth Incubator)	236
10.1.6 JEP 427: Pattern Matching for <code>switch</code> (Third Preview)	236
10.1.7 JEP 428: Structured Concurrency (Incubator)	238
10.2 Installation der Java-19-Early-Access-Builds	241
10.2.1 Allgemeine Aktionen	241
10.2.2 Weitere Aktionen unter macOS	242
10.2.3 Weitere Aktionen unter Windows	243
10.2.4 Java-19-Installation prüfen	246

11 Zusammenfassung und Schlusswort	247
11.1 Gedanken zum Umstieg	247
11.2 Fazit	248
11.3 Abschließende Hinweise	249

III Anhang	251
-------------------	------------

A Wesentliches aus Java 8, 9 und 10	253
A.1 Einstieg in Lambdas	253
A.1.1 Lambdas am Beispiel	253
A.1.2 Functional Interfaces und SAM-Typen	254
A.1.3 Type Inference und Kurzformen der Syntax	256
A.1.4 Methodenreferenzen	257
A.1.5 Das Interface <code>Predicate<T></code>	257
A.2 Streams im Überblick	259
A.2.1 Streams erzeugen – Create Operations	259
A.2.2 Intermediate und Terminal Operations im Überblick	260
A.2.3 Intermediate Operations	260
A.2.4 Terminal Operations	264
A.3 Neuerungen in der Datumsverarbeitung	265
A.3.1 Die Klassen <code>LocalDate</code> , <code>LocalTime</code> und <code>LocalDateTime</code>	265
A.3.2 Die Klasse <code>Duration</code>	267
A.3.3 Die Klasse <code>Period</code>	268
A.3.4 Datumsarithmetik mit <code>TemporalAdjusters</code>	269
A.4 Diverse Erweiterungen	271
A.4.1 Erweiterungen im Interface <code>Comparator<T></code>	271
A.4.2 Die Klasse <code>Optional<T></code>	272
A.4.3 Collection-Factory-Methoden	275
A.4.4 Erweiterungen im NIO und der Klasse <code>Files</code>	276
A.4.5 Die Klasse <code>CompletableFuture<T></code>	277
A.5 Weitere Syntaxneuerungen	280
A.5.1 Anonyme innere Klassen und der Diamond Operator	280
A.5.2 Syntaxerweiterung <code>var</code>	280
B Die Build-Tools Gradle und Maven im Überblick	283
B.1 Projektstruktur für Gradle und Maven	283
B.2 Einführung in Gradle	285
B.3 Einführung Maven	296
B.3.1 Maven im Überblick	296
B.3.2 Maven am Beispiel	300

Literaturverzeichnis **303**

Index **305**